Listing of Claims:

Claims 1-23 (Canceled)

24. (Currently Amended) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes; and

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management ~~data.~~ data; and

storing the memory management data in at least an administrative memory, the administrative memory being separate from a shared memory space.

25. (Cancelled)

26. (Original) A method according to claim 24, wherein the memory management data comprises a plurality of memory management objects associated with the plurality of processes.

27. (Original) A method according to claim 26, wherein each memory management object is uniquely associated with one of the plurality of processes.

28. (Cancelled)

29. (Original) A method system according to claim 24, wherein the set of memory blocks comprises a first memory address space.

30. (Original) A method according to claim 29, wherein the administrative memory

comprises a second memory address space.

31. (Original) A method according to claim 25, further comprising a step of each processes registering their at least one assigned memory block in the set of memory blocks in the administrative memory.

32. (Original) A method according to claim 24, further comprising a step of modifying the data in at least one assigned memory block according to the data access request.

33. (Original) A method according to claim 24, further comprising a step of returning data from at least one assigned memory block according to the data access request.

34. (Original) A method according to claim 33, further comprising a step of parsing at least one of the data access request and the returned data.

35. (Original) A method according to claim 24, further comprising a step of executing a cache module, the cache module maintaining a cache of data exchanged between the set of memory blocks.

36. (Original) A method according to claim 24, further comprising a step of assigning the at least one memory block to each process based on a buddy system allocation.

37. (Original) A method according to claim 24, further comprising a step of terminating a connection to the set of memory blocks after completion of the data access request.

38. (Original) A method according to claim 24, wherein the step of assigning at least one memory block to each of the plurality of processes comprises a step of dynamically assigning at least one memory to each of the plurality of processes.

39. (Cancelled)

40. (Cancelled)

41. (Original) A method according to claim 24, further comprising a step of copying data in one memory block to another memory block.

42. (Original) A method according to claim 24, further comprising a step of transmitting data from one memory block to a remote destination process.

43. (Original) A method according to claim 42, wherein the step of transmitting data comprises a step of transmitting data via an external network to the destination process.

Claims 44-66 (Cancelled)

67. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

modifying data in at least one assigned memory block according to the data access request.

68. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

executing a cache module, the cache module maintaining a cache of data exchanged between the set of memory blocks.

69. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

assigning the at least one memory block to each process based on a buddy system allocation.

70. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

terminating a connection to the set of memory blocks after completion of the data access request.

71. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes, wherein the step of assigning the at least one memory block comprises a step of dynamically assigning at least one memory block to each of the plurality of processes; and

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data.

72. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

copying data in one memory block to another memory block.

73. (New) A method for managing communication between processes, comprising:

assigning at least one memory block from a set of memory blocks to each of a plurality of processes;

directing a data access request from at least one of the plurality of processes to the at least one corresponding assigned memory block based on memory management data; and

transmitting data from one memory block to a remote destination process.